

Лабораторная работа № 3 по курсу : Операционные системы

Выполнил студент группы М8О-201Б-21 Кварацхелия Александр.

Условие

Ознакомиться с сигналами операционной системы UNIX/LINUX, используя утилиту `strace`, проанализировать результаты, сопоставить их с кодом программы.

Метод решения

Использовать свободно распространяемую утилиту `strace` следующим образом:
`strace lab2`

Код программы

0.0.1 main.cpp

```
#include <bits/stdc++.h>

#include "utils.hpp"
#include "body.hpp"

int main()
{
    std::vector<int> arr;

    for (int i = 0; i < 16; ++i){
        arr.push_back(GetRandomInt(0, 32));
    }

    for (int elem : arr){
        std::cout << elem << "_";
    }
    std::cout << '\n';

    TimSort(arr, 2);

    for (int elem : arr){
        std::cout << elem << "_";
    }
    std::cout << '\n';
}
```

```

    return 0;
}

```

utils.hpp

```

#ifndef UTILS_HPP
#define UTILS_HPP

```

```

#include <random>

```

```

int GetRandomInt(int min, int max);

```

```

#endif

```

body.hpp

```

#ifndef BODY_HPP
#define BODY_HPP

```

```

#include <vector>
#include <fstream>
#include <pthread.h>

```

```

#include <iostream>

```

```

void TimSort(std::vector<int> &arr, int numThreads);

```

```

#endif

```

0.0.2 utils.cpp

```

#include "utils.hpp"

```

```

int GetRandomInt(const int min, const int max)
{
    std::random_device rd;
    std::mt19937 mt(rd());
    std::uniform_int_distribution<int> dist(min, max);

    return dist(mt);
}

```

Выводы

Вызов *fork* дублирует породивший его процесс со всеми его переменными, файловыми дескрипторами, приоритетами процесса, рабочий и корневой каталоги, и сегментами выделенной памяти.

Ребёнок **не** наследует:

- идентификатора процесса (PID, PPID);
- израсходованного времени ЦП (оно обнуляется);
- сигналов процесса-родителя, требующих ответа;
- заблокированных файлов (record locking).

В процессе выполнения лабораторной работы были приобретены навыки практического применения создания, обработки и отслеживания их состояния. Для выполнения данного варианта задания создание потоков как таковых не требуется, так как всю работу выполняет системный вызов «*exes*».